

#Abyan Ardiatama

#24060120140161

#Praktikum ke-2 Machine Learning

## ##Soal 2

###Mencari dataset baru dan evaluasi menggunakan 3 algoritma yang ada di modul

###Memuat dataset ke dalam google colab menggunakan pandas dengan memasukkan url dan nama atributnya

```
import pandas
#Memuat Dataset
url =
"https://archive.ics.uci.edu/ml/machine-learning-databases/blood-
transfusion/transfusion.data"
#names = ['mpg', 'cylinders', 'displacement', 'horsepower', 'weight']
dataset = pandas.read_csv(url)

print(dataset.head(5))
```

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time
0	2	50	12500	
98				
1	0	13	3250	
28				
2	1	16	4000	
35				
3	2	20	5000	
45				
4	1	24	6000	
77				

	whether he/she donated blood in March 2007
0	1
1	1
2	1
3	1
4	0

###Membagi dataset menjadi 80% data training dan 20% data validasi

```
from sklearn import*
array = dataset.values
X = array[:,0:4]
Y = array[:,4]
validation_size = 0.20
seed = 7
```

```

X_train, X_validation, Y_train, Y_validation =
model_selection.train_test_split(X, Y, test_size=validation_size,
random_state=seed)

# Test options and evaluation metric
seed = 7
scoring = 'accuracy'

###Memvalidasi dataset pada 5 algoritma (KNN,GaussianNB,dan SVM)

from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import*
# Spot Check Algorithms
models = []
models.append(('KNN', KNeighborsClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))

# evaluate each model in turn
results = []
names = []

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=seed)
    cv_results = model_selection.cross_val_score(model, X_train,
Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    error_score='raise'
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

KNN: 0.757521 (0.021764)
NB: 0.769314 (0.049525)
SVM: 0.762619 (0.028621)

```

###Meguji keakuratan metode yang memiliki akurasi perkiraan tertinggi --> **GaussianNB**

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
# Make predictions on validation dataset
NB = GaussianNB()
NB.fit(X_train, Y_train)
predictions = NB.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

```

0.7066666666666667

[[98 14]

[30 8]]

	precision	recall	f1-score	support
0	0.77	0.88	0.82	112
1	0.36	0.21	0.27	38
accuracy			0.71	150
macro avg	0.56	0.54	0.54	150
weighted avg	0.66	0.71	0.68	150